Software Requirements and Design Document

For

Group <6>

Version 1.0

Authors:
Frankie Messina
Zach Porcoro
Raul Rodriguez
Andrew Stade
Peter Vasiljev

1. Overview (5 points)

The application is intended to give the user the ability to quickly view their progress towards completing their degree. Any outstanding degree requirements will be able to be viewed at a quick glance. With inspiration from FSU's flowchart system, this application will provide a checkpoint for those remaining classes that are required by the computer science department. The project will incorporate a user-friendly UI, with easy to navigate buttons and design that focus on accessibility all through a database on the web.

2. Functional Requirements (10 points)

1. Login (High priority): The functionality of the application requires the user to have an existing account. Authentication will be handled in a secure fashion by utilizing Firebase's API calls. The user will also be able to reset their password if needed through the use of a password reset link that is sent to a valid email address.
2. Registration (High priority): Creation of users with checks for validity (Possible use of other means like Google, Apple, etc to create new accounts)
3. Store class data for users (High priority): Using Firebase, the application will store the user's class data. The user will select which classes they have already completed after creating their account for the first time. The user will be able to update their class data as often as they would like through the UI.
4. Compare class data to CS flowchart (High priority): There will be a view, similar to the CS flowchart, that will allow the user to quickly see what classes they have and have not taken.
5. Create a schedule for the next semester: (High priority): Based on what classes the user still needs to take, it will create a schedule for them.
6. Display degree progress (High priority): The system will provide a progress bar/percentage that indicates at a glance how close the user is to meeting graduation requirements.
7. Polished User Interface (High priority): Focused on user accessibility with sleek design.
8. Professor ratings: (Medium priority): Shows the user what professors usually teach a certain class and shows their rating.
9. Inclusion of other popular majors: (Low priority): Allow our application to work with other popular majors at FSU.

3. Non-functional Requirements (10 points)

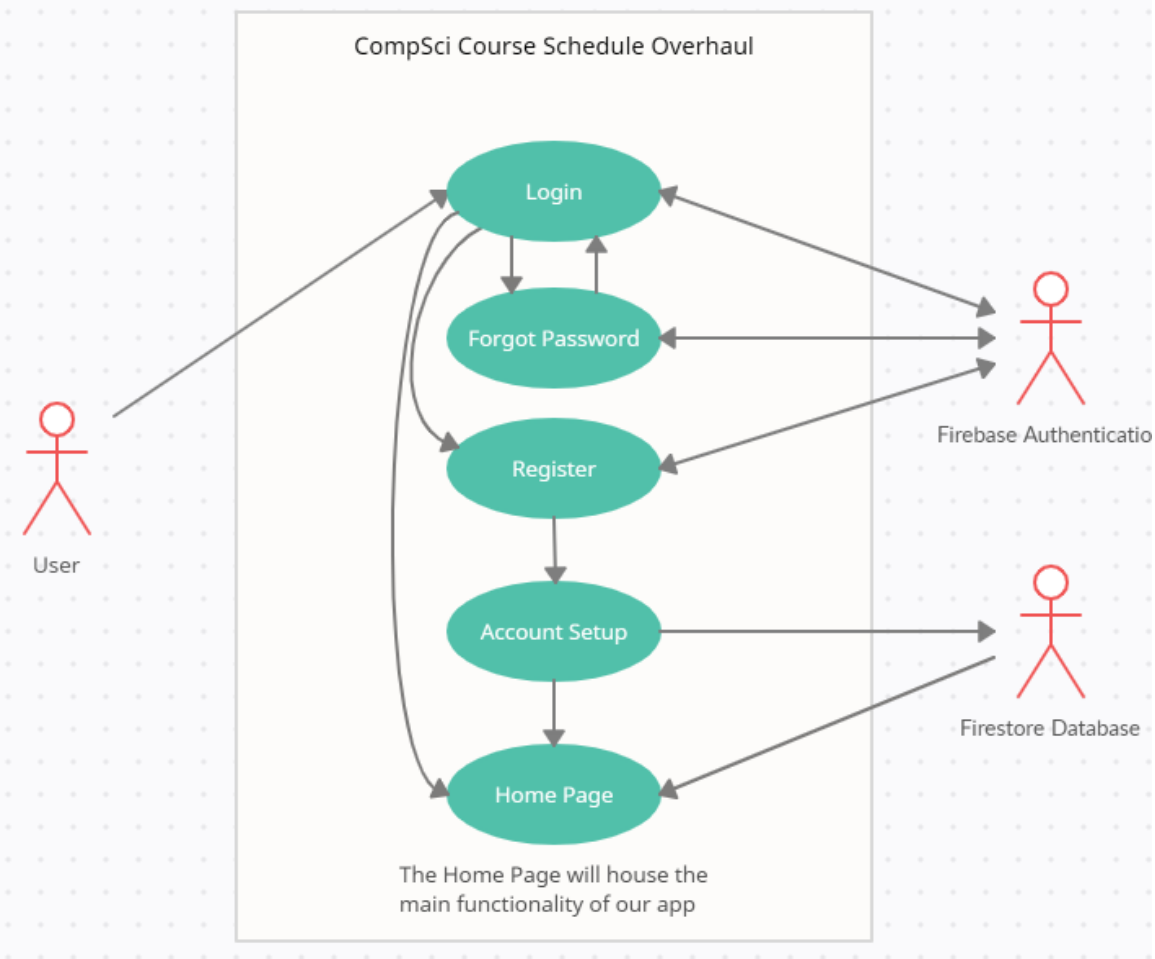
Security:
Currently, for authentication we store the user's id in local storage for the browser. However, we recognise that this is not the most secure thing to do. Going forward, we will begin to use Firebase's method of checking if the user is logged in which is much more secure. We will also go back and fix the places where we used the unsecure method.

Reliability:
We use Firebase for many different things in our project. Not only do we use it to login and create new users, but we also use it to store and retrieve data about the user's. So, if Firebase goes down, then the functionality of our application will be interrupted.

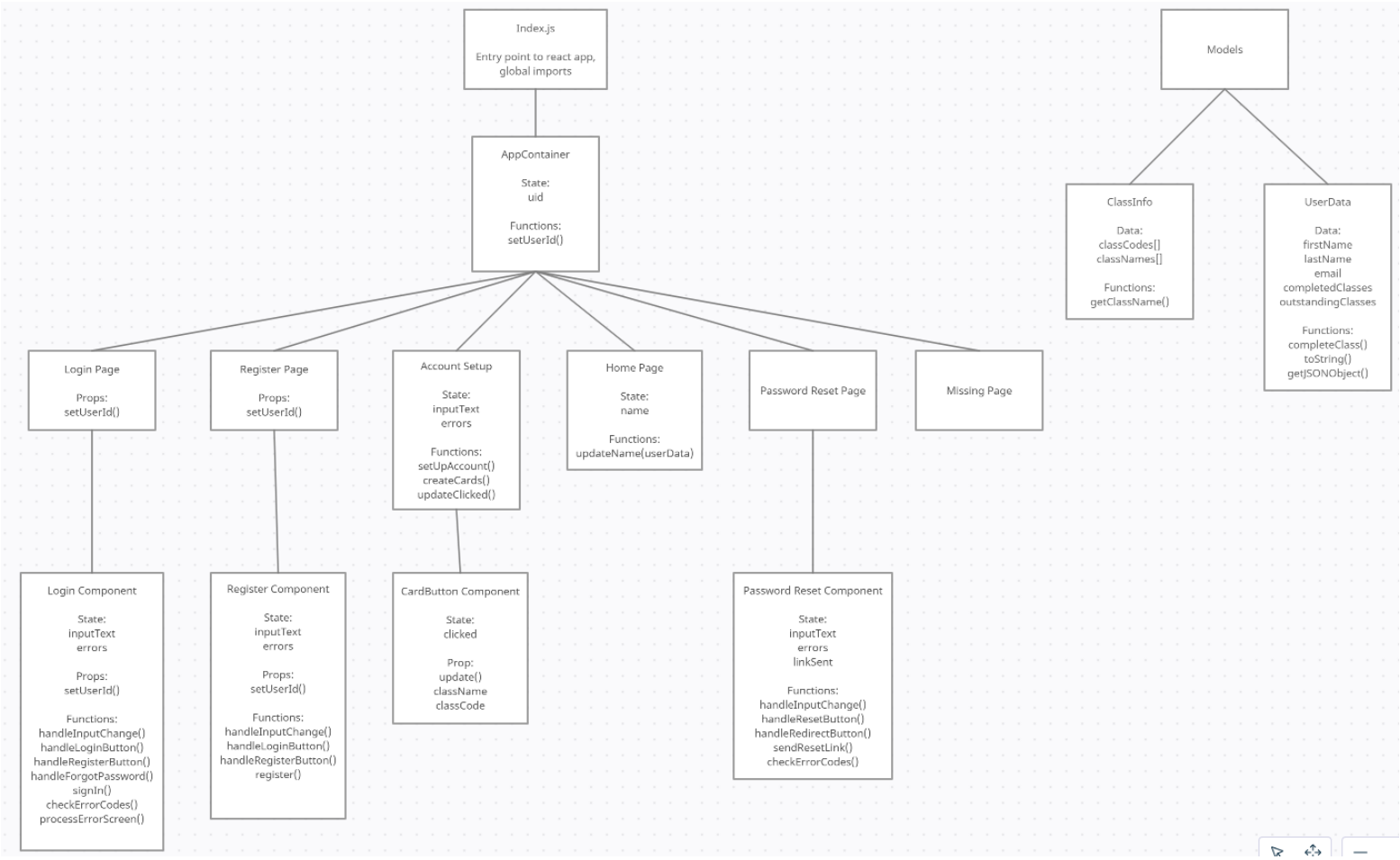
Performance:

The application must feel responsive to the user and there should not be any issues with website performance. Usage of proper Web/React development practices will help to ensure that the user has a smooth experience.

4. Use Case Diagram (10 points)



5. Class Diagram and/or Sequence Diagrams (15 points)



6. Operating Environment (5 points)

N/A for increment 1.

7. Assumptions and Dependencies (5 points)

N/A for increment 1.